



**Università degli Studi di
Trento**



Requirements Engineering meets Trust Management

P. Giorgini N. Zannone F.Massacci J. Mylopoulos

Presented by Fabio Massacci

(DIT - University of Trento - www.dit.unitn.it)

(Create-NET - www.create-net.it)



Talk Outline

- Requirements Eng. for Security & Trust
- i*/Tropos RE Methodology
- Security-Aware Tropos
 - Informal Model
 - Methodology
 - Formal Model
 - Verification
- Future Work and Conclusions



Requirements Eng. for Security & Trust I

- **UML Proposals**
 - Model-Driven Architecture [Basin et al.]
 - UMLsec - [Juriens]
- **Early Requirements Proposals**
 - Anti-requirements [van Lamsweerde et al., Crook et al.],
 - Problem-Frames [Hall et al.]
 - Security Patterns [Giorgini & Mouratidis]
 - Privacy Modelling [Liu et al., Spafford]



Requirements Eng. Security & Trust II

- **UML Pros and Cons**
 - Well-known even if meta-level extension not standardized
 - Effective - MDA -> automatic configuration of system
 - “Too Late” - model of system rather than organization
 - An average doc for “Security Policy and Security Management” (compulsory for italian public administration) is 30% on ICT systems - 70% on paper or organizational requirements
 - Worse for privacy legislation
- **Early requirements Pros and Cons**
 - Capture organizational structure
 - Modelling done at object-level (almost no extension)
 - “Too Functional” - Security is modelled explicitly and in parallel with the actual functional model



Requirements Eng. Security & Trust III

- **Where's trust?**
 - Most approaches focus on “modelling and implementing” security and privacy services
 - **The choice of Security services derives from (implicit) (dis)trust relationships linked to some functional goals**
 - Emerged if explanation for choices of security services is sought
 - Somebody must have already analyzed the requirements and found out that we do/don't trust somebody and so we need security services
 - Security&Privacy requirements are “late requirements”
- **Modelling Trust/Functional Model together**
 - In MDA spirit appropriate security services should be automatically derived from trust requirements
 - Derive Trust Management Credentials from Functional requirements + Trust/Ownership realtions



i* - Tropos Methodology I

- **Agent-Oriented RE Methodology**
 - Agents, Roles
 - Goals, Tasks, Resources
 - Dependency among agents (A depends on B on G, if A wants G to be done and B agrees to look after that)
 - Goal Decomposition (AND/OR, positive, negative contribution)
- **Easy to Understand by Users for Early RE**
- **Good for Modelling Organizations**
- **Formal Reasoning Tools Available**



i* - Tropos Methodology II

- **Limitation**

- Mindset is Cooperative Information Systems
- Only Functional Requirements, No trust at language level
- Major assumption is that if you provide a service you have also the authority to decide who can use it

- **Life is complicated...**

- Req: Data analysis of donated placenta managed by Institute's XXX IS and should be searched by authorized researcher
- Req: Decision to grant placenta cells usage for bone marrow transplant by the trusted XXX board
- Law: Genetic data belongs to the donor patient
- Law: Donor have access to (some) cells for her own family if she so required at time of donation
- Authorized by who? Trusted by who? When permissions are gathered?



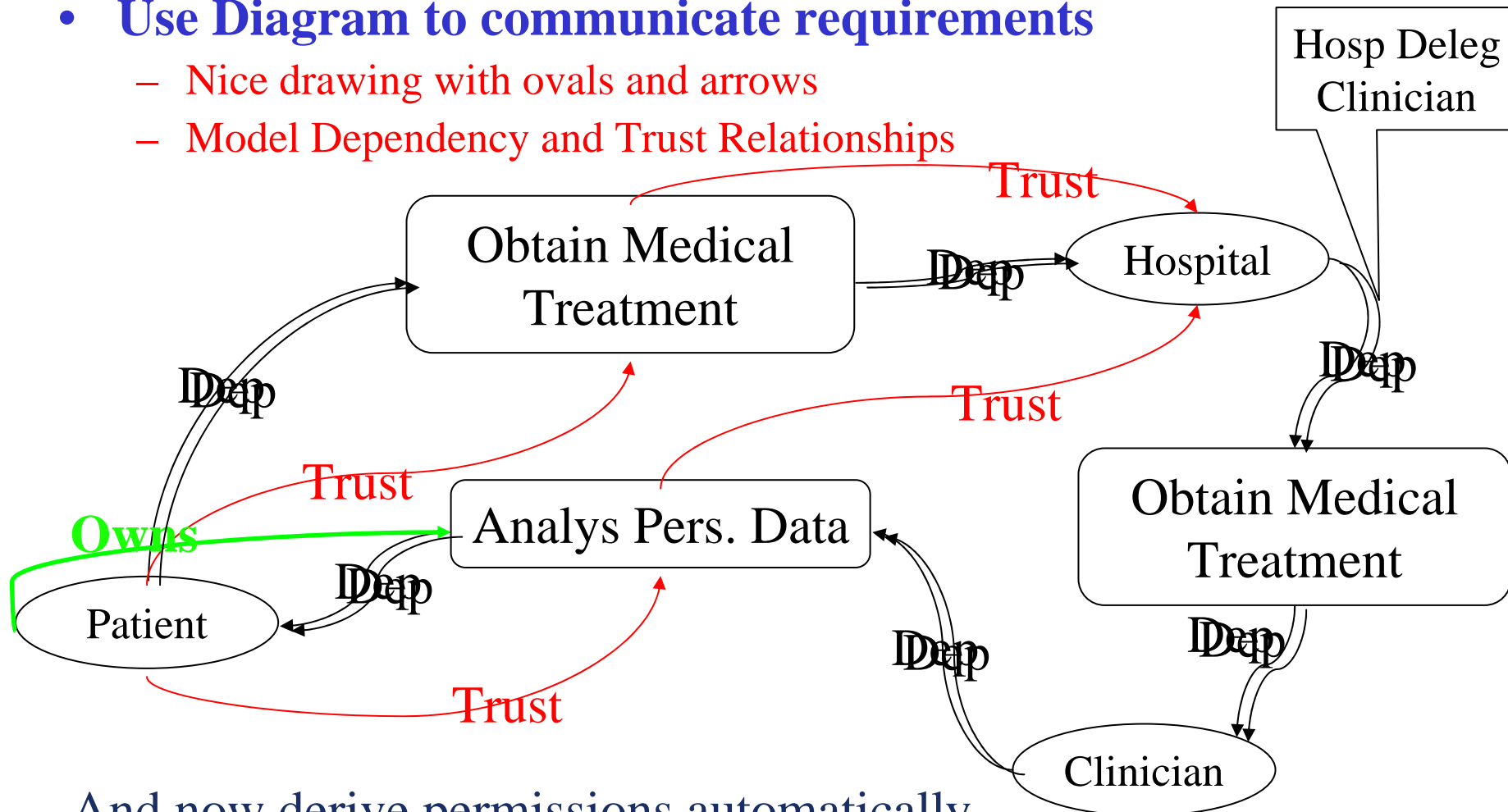
Security-Aware Tropos - Informal Model

- **Add-ons**
 - Distinction between wanting/offering/owning a goal
 - Trust relationship on Agent/goals/Agents
- **Some agents want some goal/task to be done**
 - Hospital doctor wants to consult medical record
- **Other agents offer this goal/task**
 - Nephrology ward locker stores medical records
- **Another agent owns this goal/task**
 - Patient owns the medical record
- **Agents trust other agents on the goal**
 - Patient trust Hospital to store medical record

Security-Aware Tropos - Informal II

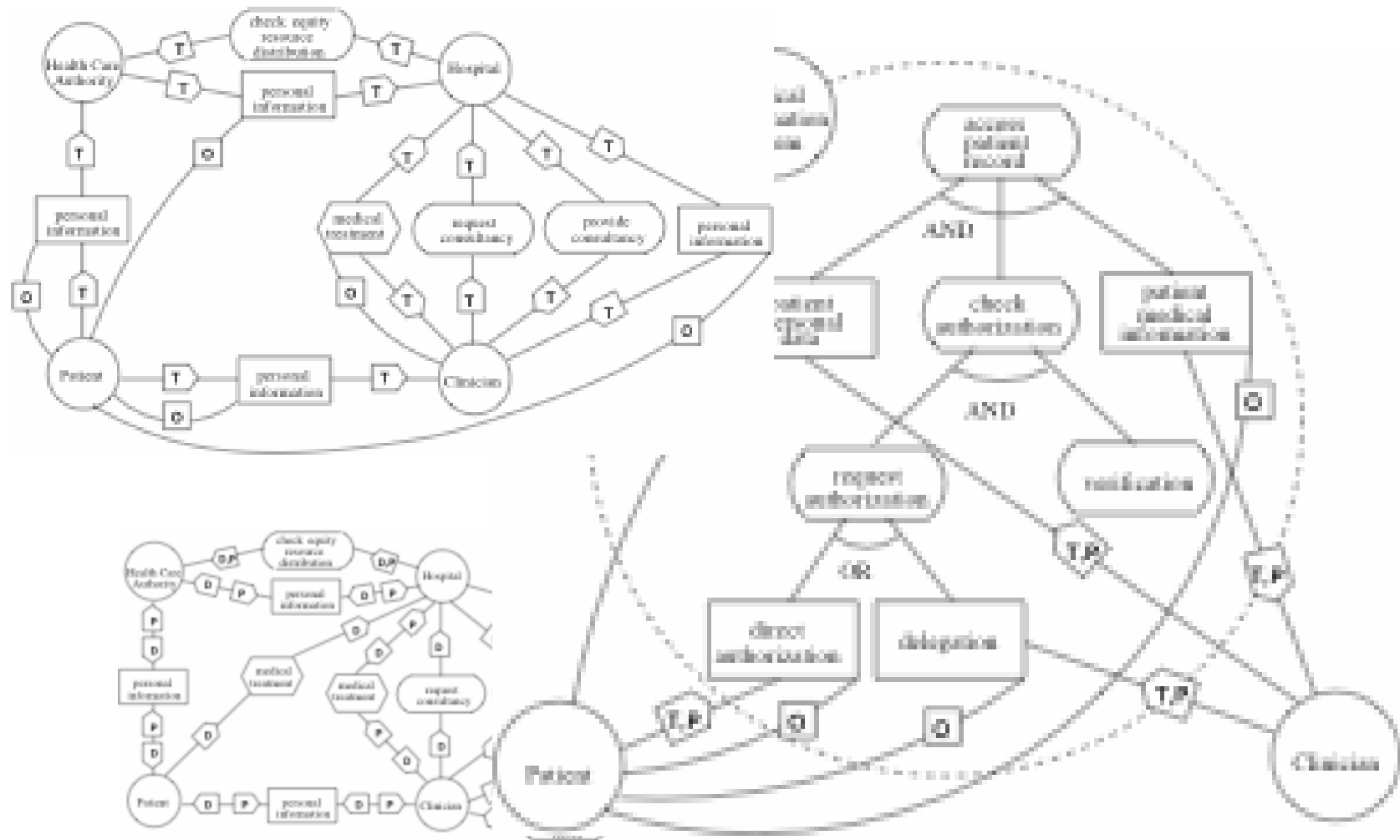
- Use Diagram to communicate requirements

- Nice drawing with ovals and arrows
- Model Dependency and Trust Relationships



And now derive permissions automatically

Security-Aware Tropos - Informal III





Security-Aware Tropos - Informal IV

- **Beware: NOT all “permissions” and “auth. processes” are mapped onto digital certificates**
- **Remember**
 - Fabio’s 30-70 “Real-Life Dominance Rule”
 - Some “credentials” will be papers signed by an individual
 - Others corresponds to oral or physical permissions
- **Examples**
 - Permission to use genetic data for research and “self” usage (collected on the day of childbirth...)
 - Parents’ or Guardian approval for clinical data to be used in case of unexpected outcome in “wet” surgery
 - University visiting professor’s request for temp. IP address



Security-Aware Tropos - Methodology

- Design Functional Dependency Model
- Design Trust/Ownership Model
- Refinements by
 - Goal Decomposition
 - Goal (Functional) Delegation to other agents
 - Modify Trust Relationship
- Design /Synth Trust Management Implementation
 - Goal (Permission) Delegation to other agents
- (Computer Supported) Analysis
 - Goal Fulfillment (Functional Delegation Chain)
 - Trusted Execution (Trust Chain Match Funct. Deleg. Chain)
 - Trusted Delegation (Trust Chain Match Permis. Deleg. Chain)



Security-Aware Tropos - Formal Model

- **Semi-formal Analysis**
 - Annotate diagrams with formulae
 - Partial checks at type level
 - Eliminate already many errors in the chains
- **Formal Analysis**
 - Full model at instance level
 - Define instances of agents and goals
 - instantiate delegation in many ways
 - Finite State Model checking and (to be done) infinite state analysis
 - Allows Discovery of subtler relationships between parties
 - Patient trusts “her” clinician, and hospital
 - Analys relationships with third parties
 - Delegation of permissions can create unexpected breach of trust
 - “natural & simple” permission chain may not match “natural” trust chain



Security-Aware Tropos -Formal Model II

- **Possible models**

- Linear Temporal Logic
- Answer-Set Programs (Datalog with constraints)
 - Used also for some Trust Management Language (RT)

- **Formulae**

- Delegation \rightarrow `delegate(Alice, Goal, Bob, Depth)`
 - Depth is for depth of re-delegation
- Ownership \rightarrow `owns(Alice, Goal)`

- **Axioms**

- `delChain(A,G,B,D) :- delegate(A,G,C,D1), D1>1, #succ(D1,D2), delChain(C,G,B,D3), D2>=D3.`
- `trustFull(hospital, Record, medicalIS) :- isRecord(Record).`
- `trustFull(hospital, Record, Agent) :- isClinician(Agent), isRecord(Record).`



Security-Aware Tropos - Reasoning

- For any delegation step there is also a trust step
- For any delegation chain is there a trust chain?
 - Not implied by the first (illegal delegation steps, or trust has different depth than actual delegation of permission)
 - hospital gives unbounded delegation to clinician to consult colleagues to consult colleagues etc.
 - Patient trust (derived from trust in hospital) stops at first consultancy.
 - Procedure should go back to patient who should get back to hospital to get additional consultancy and then get back to first consultant
- Does somebody fulfils a goal for which he has no permission?
 - Delegation of Permission does NOT start with legitimate owner of data
 - Hospital authorizes doctor but has not got patient authorization



Security-Aware Tropos - Reasoning II

- **Effective Datalog Implementation DLV (TU Wien)**
- **Generate Instance (semi-automatic)**
- **Ask queries (SQL frontend)**
 - Consistency/Inconsistency
 - Find all Agents/Goals satisfying bad property
- **Find Missing properties**
 - Abduction of missing edges in trust chain
- **Usefulness**
 - Must be grounded first, so if no error found no guarantee yet that model is correct (maybe larger instance will do)
 - Mostly useful as debugging tool for supporting Requirements Engineer



Future Works and Conclusions

- **Integrated Modelling of Trust/Functional Reqs**
 - Greater expressivity and flexibility
 - Explain “Why” and not also “How”
 - Formal reasoning possible
 - Does functional delegation chain match trust chain?
- **Future Works**
 - Automatic derivation of Trust-Management credentials for RT [EuroPKI-2004]
 - Natural way to model privacy requirements [ISPW-2004]
 - What if you must delegate (functional goals) to somebody you don't trust?
 - Integrating time: trust and reputation changes over time